一个基于迭代局部搜索的区划问题算法

孔云峰

(河南大学黄河中下游数字地理技术教育部重点实验室，河南 开封，475000)

摘要：区划问题是将特定地理区域划分为若干空间连续的分区，满足分区内差异最小和分区间差异最大这一基本原则，广泛应用于地理、环境、生态、经济、农业、城市等领域。60 余年来，学者尝试建立各种区划问题数学模型，设计了一系列的求解算法，包括精确算法、基于聚类的算法、启发式算法和基于树图的算法。针对现有算法计算效率与求解质量难以兼顾这一局限，本文提出了一个基于迭代局部搜索(ILS)的区划问题算法。该算法主要机制包括：通过邻域单元移动改进分区质量；参照中心单元快速计算分区方差提升算法速度；使用扰动机制跳出局部最优状态；更新分区中心点提升分区方案目标值；以及使用群搜索探索更大的解空间；算法各步骤中通过分区修复保持分区空间连续。55 个基准案例测试表明：ILS 算法求解质量优于 ARISEL 算法和 SKATER 算法，计算时间大幅低于 ARISEL 算法。一个多指标气候分区实验也验证了 ILS 算法的实用性。本文 ILS 算法兼顾分区质量和计算效率，并允许一个分区包含多个空间连续且面积较大的区域，具有灵活性和实用性。

关键词：区划问题；迭代局部搜索；基准测试；案例研究

1 研究背景

区划是地理学中的一个基本问题。区划是从区域角度观察和研究地域综合体, 探讨区域单元的形成发展、分异组合、划分合并和相互联系, 是对过程和类型综合研究的概括和总结(郑度等，2005)[1]。上百年来，区划理论、方法与应用研究取得了显著的进展，广泛应用于地理、环境、生态、经济、农业、城市、地图制图、空间统计等领域。随着我国社会经济的快速发展，为满足国家和地区各行业的战略决策、规划、管理和政策制定，区划仍是一个基础性的研究领域。

纵观上世纪 60 年代以来国内外区划研究进展，区划研究有两大重点和难点：区划的理论基础和分区边界的确定。前者基于地理现象的空间格局、结构、过程、机理和异质性规律，针对区划需求，确定区划目标和原则，并遴选区划指标。后者属于定量分析范畴，是在理论研究的基础上，运用地图制图、空间分析、空间聚类、空间优化等手段，科学合理地划定分区边界。刘燕华等(2005)对于编制中国综合区划方案所要解决的重要科学技术问题进行了深入分析[2]。郑度等(2008)深入阐述了自然地理区划的内涵, 提出了自然地理区划范式及关键科学问题[3]，相关思想、理论和方法对于自然地理和其他领域区划问题具有指导和借鉴价值。

区划问题(regionalization problem)是将特定地理区域划分为若干空间连续的分区，满足分区内差异最小化和分区间差异最大化这一基本原则，本质上是一个增加了分区空间连续约束的聚类问题。聚类问题本身的计算复杂度极高，分区空间连续约束使得区划问题求解更为困难。自上世纪 60 年代开始，学者尝试建立各种数学模型，并设计了一系列算法，包

基金项目：国家自然科学基金项目（41871307）。[Foundation item: National Natural Science Foundation of China, No. 41871307.]

*作者：孔云峰（1967— ），男，河南洛阳人，教授，主要从事空间分析、空间优化等研究。E-mail: yfkong@henu.edu.cn

括精确算法、基于聚类的算法、元启发式算法、树图分割算法和混合启发算法(Duque et al., 2007)[4]。然而，这些算法存在以下局限：精确算法计算复杂度过高；经典聚类算法难以处理分区空间连续性；树图分割算法仅依据相邻单元生成树图，分割后难以保证区划质量；元启发式算法求解质量较好，但计算时间过长。针对现有区划问题算法这些局限，本文提出一个新的区划算法，既要满足区划质量，又能够降低算法复杂度。

2 文献回顾

区划问题也称为空间分类问题(spatial classification problem)、空间聚类问题(spatial clustering problem)、空间聚合问题(spatial aggregation problem)、空间分区问题(spatial districting problem)、分区设计问题(zone design problem)等。尽管这些术语有差别，但内涵基本一致，均致力于将地理空间划分为若干区域，在满足特定约束条件的前提下确定最优的分区方案。

区划问题数学建模和模型求解的复杂性主要体现在分区空间连续这一约束条件。针对各种区划需求，不少学者详细定义了区划决策变量、约束条件和目标函数，提出了多个区划问题数学模型(Cliff et al., 1975; Keane, 1975; Wright et al., 1983; Cova and Church, 2000; Williams, 2002; Shirabe, 2005; Duque et al., 2011; Li et al., 2014)[5]。Keane(1975) [6]证明了空间连续性约束区划问题是一类 NP-Hard 问题。因此，区划问题模型求解的计算复杂度极高。*P*-regions 问题将 n 个空间单元划分为 p 个连续区域，是一个经典的区划问题。该问题可表达为三种混合整型规划(MIP)模型：树模型、次序模型和网络流模型(Duque et al., 2011)[11]。然而，基于数学模型的精确算法仅能够求解空间单元数量很少的小规模问题。例如，CPLEX 模型计算表明：针对 n=49 和 p=3~10 的基准案例，3 小时的计算时间均不能获得最优解[11]。

基于聚类分析的区划方法包括：基于经典聚类分析的方法、距离加权聚类分析方法、M-means 聚类方法和凝聚层次聚类方法。前三种方法，思路简单，但处理空间连续性分区的能力不足，以牺牲分区质量为代价保证分区空间连续。经典的层次聚类方法较为成功地应用于区划，算法流程如下：(1)首先将每一个空间单元作为一个分区；(2)计算各分区见的相似度；(3)寻找近似度最接近且连续的两个分区，把他们合并为一个分区；(4)重复步骤(2)和(3)，直到分区数量满足区划目标。步骤(2)中分区相似度的计算有多种方法，如方差最小(Ward)、两个分区中最接近单元的相似度(single linkage)、两个分区中两个差异最大单元的相似度(complete linkage)、两个分区中单元均值或中值的相似度(average linkage)等。步骤(3)限制相邻区域合并，保证分区的连续性。该方法采用自下而上的分区合并策略，适合于分区数量不确定的区划问题，但步骤(2)相似度计算方法和步骤(3)空间邻接约束对于聚类树的形成影响很大(Guo, 2008)[13]。

启发式区划算法的基本原理是：先构造一个可行的区划方案，再使用邻域算子进行迭代搜索改进。第一步使用区域种子生长、聚类分析或其他简单算法构造一个可行的区划解，第二步根据当前的分区方案和空间单元间的空间关系，尝试进行空间单元的移动迭代地改

进区划方案。AZP 方法是 Openshaw (1977)[14]提出的一个经典区划算法，先将 n 个空间单元随机划分为 k 个区域，在顾及分区空间连续约束的前提下，尝试将某个单元重新分配到另一个区域，持续改进区划方案。本质上，该算法属于爬山算法，搜索过程容易陷入局部最优而过早停滞。

为避免邻域搜索过程陷入局部最优，学者不断改进算法，通过模拟退火(Browdy, 1990; Openshaw and Rao, 1995)[15][16]、禁忌(Openshaw and Rao, 1995)[16]等元启发机制，提升搜索过程的多样性，从而获得较高质量的区划方案。Duque and Church(2004)改进禁忌算法为 ARISEL 算法[17]。该算法使用简单方法提供多个初始区划方案，选择高质量区划方案进行禁忌搜索。

为降低邻域搜索的计算复杂度，学者提出了基于树图的启发式算法：先将区域抽象为网络图，再将网络图简化为树图，通过树分割获得空间连续的区域。树结点代表空间单元，树干表示空间单元间的邻接关系(郭仁忠,1985)[18]。Maravalle and Simeone(1995) [19]提出了一个基于树图的区划问题算法(MIDAS)：根据图 G 生成树 T，删除 T 的 p-1 条连接获得 p 棵子树，代表 p 个空间连续的区域。因树 T 上的解空间很有限，MIDAS 算法尝试不断调整树 T 为 T*获得更好的分区方案。此后，Assunção et al. (2006)基于最小生成树概念提出了一个区划问题算法 SKATER[20]。Guo(2008)改进该算法为 RADCAP 算法，提出了 6 种动态树生成方法[13]：First-Order-SLK、First-Order-CLK、First-Order-ALK、Full-Order-SLK、Full-Order-CLK 和 Full-Order-ALK。实验发现：Full-Order-CLK 和 Full-Order-ALK 优于其他生成树方法。

总体上，聚类算法思路简单且容易实现，但这类方法有的难以保证分区连续性，有的虽顾及分区连续性但不能保证全局优化质量。启发式算法类型众多，启发改进方法设计思路简单，但优化性能有限；元启发方法性能较高，但这一类算法设计较为复杂，计算效率偏低。基于生成树的方法，计算效率大幅提升，但同时大幅降低了搜索空间，影响到区划质量。Aydin et al. (2021)[21]设计了基准测试案例，测试了 AZP、AZP-SA、AZP-Tabu、ARISEL、SKATER 和 REDCAP 算法。计算结果表明：ARISEL 算法总体质量最高，但计算速度很慢；SKATER 算法求解质量尚好，计算效率非常高。考虑到区划问题应用领域越来越广，在应用中对于区域规划、决策影响较大，有必要设计更有效地的区划算法，既保证区划质量，又能够快速计算。

3 区划问题定义

某一地理区域共有 n 个空间单元，记为集合 U={1, 2, 3 … n}。每个单元有 m 个属性，记为集合 A={1, 2, 3 … m}，单元 i 属性为 $a_{i1}, a_{i2}, a_{i3} \dots a_{im}$。将地理区域划分为 p 个空间连续的分区，记为集合 C={1, 2, 3 … p}，分区 i 包含的地理单元解为 c_i，满足$c_i \cap c_j = \phi(i \neq j)$和 $c_1 \cup c_2 \cup c_3 \cup \dots c_p = U$，即任意两个分区无重叠，每个空间单元必须划分在特定分区中。分区目标是分区内单元属性方差之和最小：

$$f(C) = \sum_{i \in C} \sum_{j \in c_i} \sum_{k \in A} \left(a_{jk} - \bar{a}_{ik}\right)^2 \quad (1)$$

式(1)中$\bar{a}_{ik}$为分区 i 中所有单元属性 k 的平均值。

在区划实践中，有几个实际问题需要考虑。首先，考虑到不同属性的含义和量纲存在差异，通常采用标准化后的单元属性值。常用的数据标准化方法包括：标准差标准化方法、最大最小极值标准化方法、线性比例标准化方法等。其次，考虑属性的重要性可能不同，可为每个属性设置权重。令单元 i 标准化属性值为 $b_{i1}, b_{i2}, b_{i3} \ldots b_{im}$ 为标准化属性值，属性 k 的权重为w_k，区划目标函数为：

$$f(C) = \sum_{i\in C}\sum_{j\in c_i}\sum_{k\in A} w_k\left(b_{jk} - \bar{b}_{ik}\right)^2 \tag{2}$$

一般地，可统计每个属性的R^2指标评价分区质量：

$$R_k^2 = 1 - \sum_{i\in C}\sum_{j\in c_i} w_k\left(b_{jk} - \bar{b}_{ik}\right)^2 / \sum_{j\in U} w_k\left(b_{jk} - \bar{b}_k\right)^2 \tag{3}$$

其中$\bar{b}_k$为属性 k 的平均值。如采用标准差标准化方法，则均值$\bar{b}_k = 0$。同时，可计算总体R^2指标评价分区质量：

$$R^2 = 1 - \sum_{i\in C}\sum_{j\in c_i}\sum_{k\in A} w_k\left(b_{jk} - \bar{b}_{ik}\right)^2 / \sum_{j\in \mathrm{U}}\sum_{k\in A} w_k\left(b_{jk} - \bar{b}_k\right)^2 \tag{4}$$

4 算法设计

求解区划问题存在多个难点。首先，本文将区划问题定义为一个增加了空间连续约束的聚类问题，满足空间连续条件使区划问题求解变得较为复杂。区划算法中，分区空间连续判断与空间连续修复是求解算法中两个较为频繁的操作。其次，目标函数(2)中分区 i 的均值$\bar{b}_{ik}$计算较为复杂，容易造成算法计算效率偏低。为加快计算，可采用分区中心点的属性值取代均值$\bar{b}_{ik}$。在区划方案中，确定每个分区的中心点，有利于快速评估区划方案目标值。基于以上分析，本文采用基于中心的区划算法，并保持每个分区空间连续。

选择迭代局部搜索(ILS)算法作为求解区划问题的算法框架。ILS 算法思路简单、易于实现，对于离散优化问题行之有效(Lourenço et al., 2010)[22]。该算法从一个初始解开始，迭代地进行扰动和局部搜索。局部搜索容易陷入局部最优，对当前位置的扰动能够使算法脱离局部最优。初始解生成、局部搜索和扰动使 ILS 算法的基本模块。为提升 ILS 算法优化性能，本文改造单解 ILS 算法为群解 ILS 算法。改进 ILS 算法流程如下：

```
参数: 群大小 (psize)，破坏强度(strength)，连续未更新最好解循环数(mloops)。
Pop = GenerateInitialSolutions(psize);
s_best=Best(Pop);
notImpr=0;
While notImpr < mloops:
    Select a solution s from P randomly;
    s′=Perturbation (s, strength);
    s″=LocalSearch(s′);
    s*=updateCenetrs(s″)
    If f(s*)<f(s_best): s_best=s*, notImpr=0;
        else: notImpr+=1;
    P =UpdatePopulation (P, s*);
Output s.
```

步骤(1)算法采用经典 *K*-medoids 算法产生初始解。该算法随机选择 p 个空间单元作为分区中心，迭代进行单元指派和中心点更新，直到所有中心点不能更新为止。指派是将每个空间单元指派到最近的中心单元，计算简单；受研究区形状和地理要素空间分布的影响，指派形成的分区不能满保证其空间连续，为此算法需要判断分区的空间连续性，并进行连续性修复。

步骤(7)采用分区边界单元移动方法进行局部搜索。该方法尝试移动某一个边界单元到相邻的分区，若该移动能够减少区划目标，则更新当前解。这一操作需要考虑分区的空间连续性，保证单元移动后分区连续性约束仍然得到满足。

步骤(6)进行分区方案扰动。常用的扰动方法很多，例如，破坏若干分区、破坏一个连续区域、破坏一定比例的边界单元，然后进行解的修复。若修复后，分区不能保证空间连续，则继续进行空间连续修复。

与基于单解的搜索算法相比，改进 ILS 算法维护一组解。首先，算法步骤(1)生成一组初始解；其次，每一次迭代开始，从群解中随机选择一个解作为当前解进行搜索(步骤 5)；第三，搜索完成后，使用新解更新群解(步骤 11)。群解更新中，优先考虑解的目标值，其次考虑解的差异程度，保持群解之间有一定的差异。基于群解的 ILS 算法维护一组具有差异度的精英解，扩大了解空间搜索范围，有利于改进求解质量；同时，算法收敛速度通常会变慢，计算时间有一定的增加。

因本文算法基于中心单元评估分区目标，局部搜索完成后，步骤(8)尝试更新中心单元，使分区目标值进一步降低。

空间连续性判断是区划算法中的一个关键步骤。本文使用生成树判断分区的连续性(Xiao 2008; Liu et al. 2016)[23][24]。若一个分区中的所有单元能构成一个生成树，则该分区连续。考虑到一些特殊情形，本文允许一个分区包含两个或多个面积较大的区域。图 1 左图中，蓝色、棕色和绿色分区均包括两个部分。所有蓝色或棕色部分的面积均较大，可认为蓝色区域和棕色区域是空间连续的分区。而绿色部分中，因其中一块过小，将其视为空间不连续的破碎单元。本文空间连续判断方法如下：(1)针对某一分区，从任意单元开始构造生成树；(2)若有某些单元不能连接到生成树上，则针对剩余单元构造新的生成树；(3)重复步骤(2)直到没有剩余单元；(4)计算每一个生成树对应单元数和面积，若有某一生成树单元数或面积小于规定的阈值，则认为该区域不连续；同时，单元数或面积过小生成树对应的单元是破碎单元。针对空间不连续分区，需要对破碎的单元进行修复，将其指派到最近的相邻分区中。左图中的绿色部分，有一块很小的斑块，共有 5 个单元，可将其作为破碎单元；可将其指派到邻近蓝色分区，修复后的分区如右图。

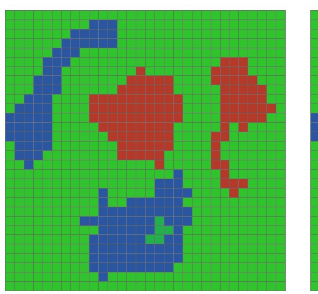
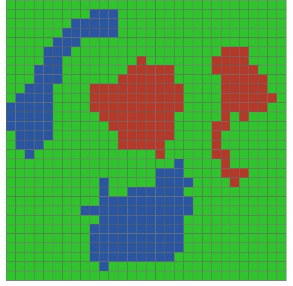



Fig. 1 Illustration of the spatially contiguous regions

图 1 分区空间连续性判断示意图

5 算法测试

5.1 基准案例测试

算法测试使用文献(Aydin et al., 2021)[21]提供的基准测试案例集。该案例集基于 3 个规则格网地图数据生成，网格数量分别为 120(10×12)、300(15×20)和 1200(30×40)。将这些地图事先划分为若干分区，并根据分区模拟每一个单元的属性数值。使用最终的模拟数据进行区划，测试算法的性能。基于每幅地图生成 18 个案例，即 2 类分区形状、3 个分区数量和 3 个数值模拟参数的组合。分区形状为简单矩形为主(A)和较复杂图形(B)，分区数量为 5、10 和 15，相邻分区属性均值差异参数设置为 2、3 和 4。另外，针对一幅 900(30×30)网格地图，模拟了形状不规则分区案例，共 5 个分区，数值模拟参数为 3。综上，共生成 55 个分区案例，每个案例的单元属性值分别随机模拟 100 次。模拟方法是：为每个分区设置一个属性均值，使用参数 2、3 或 4 设置相邻分区属性均值，以方差为 1 的正态分布随机模拟单元属性值。模拟案例基本情况见表 1，案例详细介绍见文献 Aydin et al.(2021) [21]，数据下载地址为 https://doi.org/10.6084/m9.figshare.14067239。该地址还提供了 6 种算法计算结果、质量评价指标和计算时间。

表 1 基准测试案例特征

Tab.1 Characteristics of The benchmark instances

地图名称	网格大小	分区形状	分区数量	数值模拟参数	产生区划方案数量	数值模拟次数
G120	10×12	A, B	5, 10, 15	2, 3, 4	18	100
G300	15×20	A, B	5, 10, 15	2, 3, 4	18	100
G1200	30×40	A, B	5, 10, 15	2, 3, 4	18	100
Blob	30×30	不规则	4	3	1	100

为直观地理解案例，图 2 展示了 4 种区划方案：G120-5A、G300-10B、G1200-15A 和 Blob。方案名称由地图名称、分区数量和分区形状组成。图 3 为这四个区划方案分别使用模拟参数 4、2、3 和 3 生成的模拟数值，色彩深浅代表数值的大小。模拟参数越大，不同分区单元的数值差异越大，相对容易区分事先设置的区域；反之，较难辨认事先设置的区域。

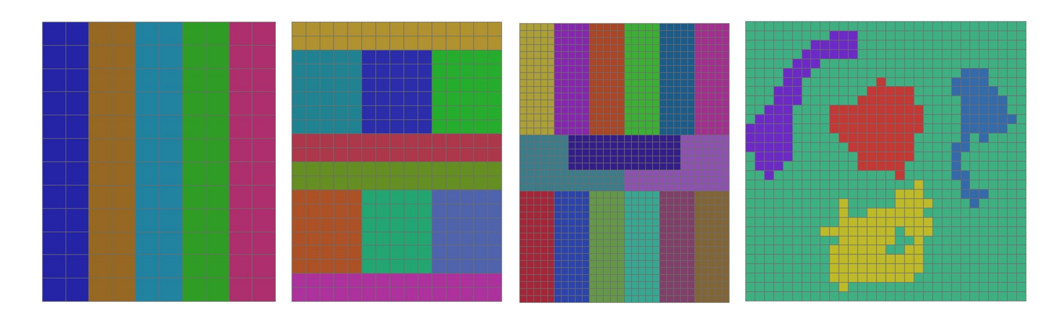


Fig. 2 Illustration of spatial units and regions (G120_5A, G300_10B, 1200_15A and BLOB)

图 2 分区示意图 (G120_5A、G300_10B、1200_15A 和 BLOB)

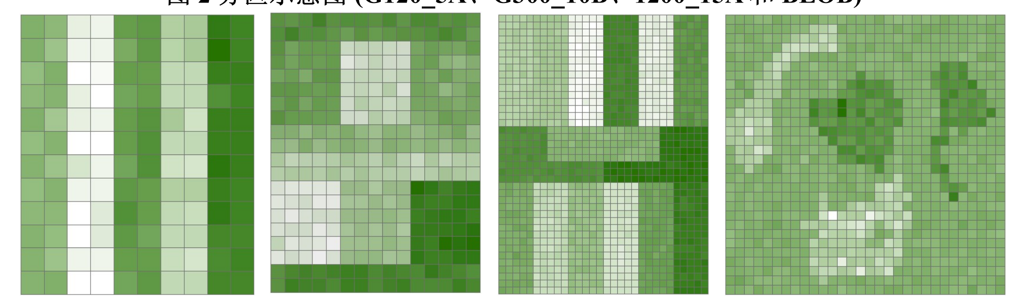


Fig. 3 Simulated values of spatial units (G120_5A4, G300_10B2, 1200_15A3 and Blob3)

图 3 单元属性数值模拟示意图 (G120_5A4、G300_10B2、1200_15A3 和 Blob3)

针对每个分区方案的 100 个数值模拟，使用本文 ILS 算法进行计算，获得 100 个区划方案，并计算每个区划方案的 ARI 指标和 R^2 指标。ARI 指标表示分区方案与真实分区的相似程度，越接近于 1 越好；R^2 指标度量分区内单元属性方差的相对大小，越接近于 1 越好。表 2 列出每个分区方案 100 个分区结果 ARI 指标和 R^2 指标的平均值，并与 Skater 和 ARISEL 算法进行比较，其中，ARISEL 和 Skater 算法区划指标来自文献[21] (Aydin et al., 2021)。可以看出：总体上 ILS 算法结果优于 ARISEL 算法，而 ARISEL 算法优于 Skater 算法。针对模拟参数为 2 的高难度案例，ILS 算法优势更为显著。

表 2 基准案例 ARI 指数和 R^2 指数均值统计

Tab. 2 ARI and R_2 indexes from 55 benchmark instances

	100 个 ARI 指数均值			100 个 R^2 指数均值		
案例名称	ILS	Skater	ARISEL	ILS	Skater	ARISEL
G120_5A2	0.804	0.665	0.717	0.901	0.870	0.901
G120_5A3	0.944	0.795	0.916	0.952	0.915	0.952
G120_5A4	0.979	0.912	0.972	0.971	0.953	0.971
G120_5B2	0.861	0.748	0.735	0.903	0.902	0.912
G120_5B3	0.959	0.884	0.927	0.952	0.949	0.953
G120_5B4	0.987	0.924	0.979	0.971	0.966	0.971
G120_10A2	0.852	0.801	0.776	0.972	0.974	0.973
G120_10A3	0.963	0.882	0.885	0.988	0.985	0.986
G120_10A4	0.987	0.926	0.953	0.993	0.991	0.992
G120_10B2	0.851	0.765	0.756	0.974	0.974	0.973
G120_10B3	0.963	0.884	0.890	0.988	0.986	0.987
G120_10B4	0.991	0.934	0.954	0.993	0.992	0.992
G120_15A2	0.850	0.806	0.765	0.986	0.989	0.986
G120_15A3	0.949	0.889	0.868	0.994	0.994	0.993
G120_15A4	0.986	0.931	0.923	0.997	0.996	0.996

G120_15B2	0.850	0.806	0.779	0.988	0.988	0.985
G120_15B3	0.937	0.881	0.867	0.994	0.993	0.992
G120_15B4	0.987	0.916	0.924	0.997	0.996	0.996
G300_5A2	0.898	0.739	0.816	0.897	0.861	0.900
G300_5A3	0.969	0.891	0.956	0.950	0.926	0.950
G300_5A4	0.991	0.944	0.988	0.971	0.959	0.971
G300_5B2	0.901	0.817	0.798	0.898	0.892	0.903
G300_5B3	0.968	0.918	0.952	0.950	0.946	0.950
G300_5B4	0.991	0.955	0.988	0.971	0.966	0.971
G300_10A2	0.869	0.759	0.804	0.968	0.950	0.961
G300_10A3	0.962	0.844	0.891	0.987	0.973	0.982
G300_10A4	0.991	0.873	0.961	0.993	0.976	0.991
G300_10B2	0.867	0.832	0.785	0.970	0.969	0.968
G300_10B3	0.973	0.915	0.897	0.988	0.985	0.985
G300_10B4	0.994	0.917	0.975	0.993	0.988	0.992
G300_15A2	0.859	0.835	0.784	0.985	0.985	0.981
G300_15A3	0.962	0.894	0.869	0.994	0.992	0.991
G300_15A4	0.991	0.924	0.933	0.997	0.995	0.995
G300_15B2	0.877	0.847	0.795	0.984	0.987	0.985
G300_15B3	0.968	0.899	0.874	0.994	0.993	0.992
G300_15B4	0.993	0.919	0.924	0.997	0.995	0.996
G1200_5A2	0.960	0.795	0.902	0.893	0.854	0.891
G1200_5A3	0.986	0.907	0.982	0.949	0.922	0.949
G1200_5A4	0.995	0.941	0.994	0.970	0.945	0.970
G1200_5B2	0.955	0.897	0.888	0.892	0.885	0.894
G1200_5B3	0.988	0.970	0.976	0.949	0.947	0.948
G1200_5B4	0.996	0.985	0.995	0.970	0.969	0.970
G1200_10A2	0.886	0.784	0.826	0.962	0.947	0.956
G1200_10A3	0.981	0.853	0.926	0.987	0.968	0.980
G1200_10A4	0.995	0.870	0.968	0.993	0.973	0.990
G1200_10B2	0.909	0.901	0.843	0.968	0.969	0.968
G1200_10B3	0.986	0.950	0.909	0.987	0.985	0.984
G1200_10B4	0.997	0.962	0.963	0.993	0.991	0.991
G1200_15A2	0.874	0.881	0.809	0.983	0.985	0.981
G1200_15A3	0.968	0.915	0.896	0.993	0.992	0.991
G1200_15A4	0.993	0.932	0.932	0.996	0.995	0.995
G1200_15B2	0.892	0.915	0.810	0.983	0.986	0.982
G1200_15B3	0.978	0.932	0.891	0.993	0.993	0.992
G1200_15B4	0.993	0.939	0.933	0.996	0.995	0.995
Blob	0.941	0.936	0.898	0.870	0.845	0.865

针对图 3 中 Blob 模拟数值案例，本文 ILS 算法的优势更加明显。图 4 中，左边 2 个区划方案为 ILS 算法结果，右边 2 个区划方案为 ArcGIS 10.3 中 Skater 算法结果。可以看出，ILS 算法较完美地还原了事先设定的分区，而 Skater 算法混淆了部分分区。ArcGIS 的计算时间约为 2.5~2.7s，ILS 算法需 5.1~7.1s。图 4 中，4 个区划指标的值分别为 0.872、0.891、0.567 和 0.719.

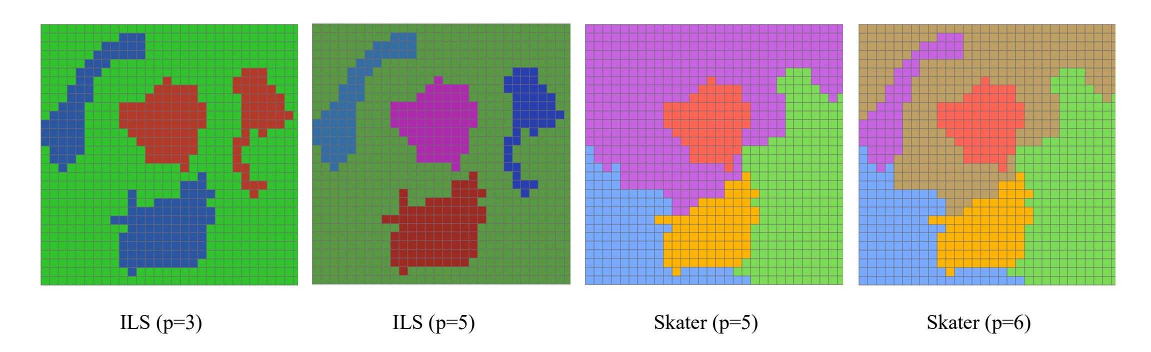


Fig. 4 Regionalization results from Blob instance

图 4 Blob 案例分区结果

表 3 为三个算法的计算时间比较，其中，ARISEL 和 Skater 算法计算时间来自文献[21] (Aydin et al., 2021)。可以看出：(1)在 ArcGIS 中实现的 Skater 算法计算速度最快，随案例规模增大，计算时间增加较少；(2)ARISEL 算法计算时间最长，随案例规模增大，计算时间快速增长；(3)本文 ILS 算法计算时间高于 Skater 算法，但远低于 ARISEL 算法。应当注意，3 个算法的计算环境差异很大，不能直接比较计算时间，但计算时间能够大体上反映各个算法的计算效率。

表 3 计算时间比较

Tab. 3 Comparison of the computation times

案例	ARISEL	Skater	ILS	案例	ARISEL	Skater	ILS
G120_05A	4.81	0.53	0.78	G120_05B	4.25	0.45	0.80
G120_10A	3.21	0.54	0.84	G120_10B	3.28	0.51	0.78
G120_15A	4.28	0.46	0.74	G120_15B	4.26	0.50	0.76
G300_05A	42.76	0.55	1.72	G300_05B	48.30	0.60	1.60
G300_10A	25.22	0.56	1.48	G300_10B	21.28	0.58	1.01
G300_15A	28.30	0.59	1.32	G300_15B	26.24	0.60	1.55
G1200_5A	1296.96	0.92	10.22	G1200_5B	1123.03	0.84	6.24
G1200_10A	740.46	0.94	7.16	G1200_10B	508.20	0.95	7.39
G1200_15A	481.21	0.95	5.49	G1200_15B	338.53	0.95	4.78

5.2 黄淮海地区气候分区

为进一步测试本文算法，选择黄淮海地区尝试进行气候区划。首先，本文黄淮海地区包括黄河、淮河、海河流域及山东半岛；其次，使用该区域 15 分网格 30 年年均降雨量和年均温度进行分区。该案例区包括 2478 个空间单元，空间范围如图 5 所示。研究区每个单元有 60 个属性数据，即 30 个年均降雨量数据和 30 个年均气温数据。

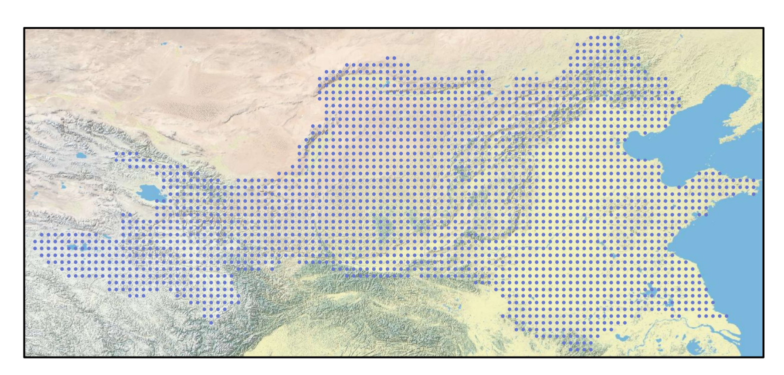


Fig. 5 Map of a case study area

图 5 研究区示意图

针对黄淮海地区气候数据，分别采用 ArcGIS 10.3 中 Skater 算法和本文 ILS 进行气候区划。气候分区数量分别设置为 3、4、5、6、7、8、9、10、12 和 15，并假定所有属性的权重相同，均为 1。Skater 算法和本文 ILS 算法均使用标准差标准化方法将 60 个属性数据进行标准化处理，完成区划后，统计每一个属性数据的 R^2 指标和标准化数据的总体 R^2 指标。表 4 中提供了 Skater 算法和 ILS 算法区划结果的 R^2 指标，包括 60 个属性 R^2 指标的最小值(MinR2)、平均值(AvgR2)和最大值(MaxR2)，也包括标准化数据的总体 R^2 指标(R2)和计算时间。可以看出，ILS 算法区划质量指标显著高于 Skater 算法，同时 Skater 算法计算效率大幅领先于 ILS 算法。

表 4 研究区气候区划 R^2 指标统计

Tab 4. The R^2 indexes from the case study area

	Skater					改进 ILS				
K	MinR2	AvgR2	MaxR2	R2	时间/s	MinR2	AvgR2	MaxR2	R2	时间/s
3	0.540	0.671	0.786	0.671	8.68	0.304	0.687	0.872	0.687	36.52
4	0.544	0.755	0.842	0.755	3.79	0.514	0.796	0.892	0.796	37.93
5	0.582	0.780	0.863	0.780	3.89	0.611	0.829	0.897	0.829	34.87
6	0.592	0.804	0.896	0.804	4.43	0.645	0.855	0.912	0.855	33.18
7	0.604	0.826	0.899	0.826	4.55	0.635	0.869	0.927	0.869	37.75
8	0.621	0.846	0.924	0.846	4.55	0.692	0.885	0.943	0.885	31.58
9	0.692	0.862	0.926	0.862	4.62	0.706	0.894	0.945	0.894	32.10
10	0.717	0.871	0.926	0.871	4.60	0.743	0.905	0.951	0.905	47.18
12	0.732	0.888	0.937	0.888	4.77	0.779	0.917	0.962	0.917	46.62
15	0.752	0.908	0.950	0.908	4.99	0.810	0.930	0.968	0.930	35.40

图 6 为分区数量为 6 时，Skater 算法和 ILS 算法的区划结果。可以看出，两者区划结果差异较大，表现在区域形状、大小和边界的差异。Skater 算法结果中，分区呈块状；而 ILS 算法结果中，分区更倾向于条带形状，符合该研究区气温、降雨量、地形变化的总体空间规律。从 R^2 指标看，ILS 算法分区指标(0.855)显著优于 Skater 分区指标(0.804)。Skater 算法基于相邻单元的相似性，不考虑非邻单元之间的关系，导致算法具有一定的局限性。ILS 算法克服了 Skater 算法过于关注局部的局限，从而使提升了区划质量。

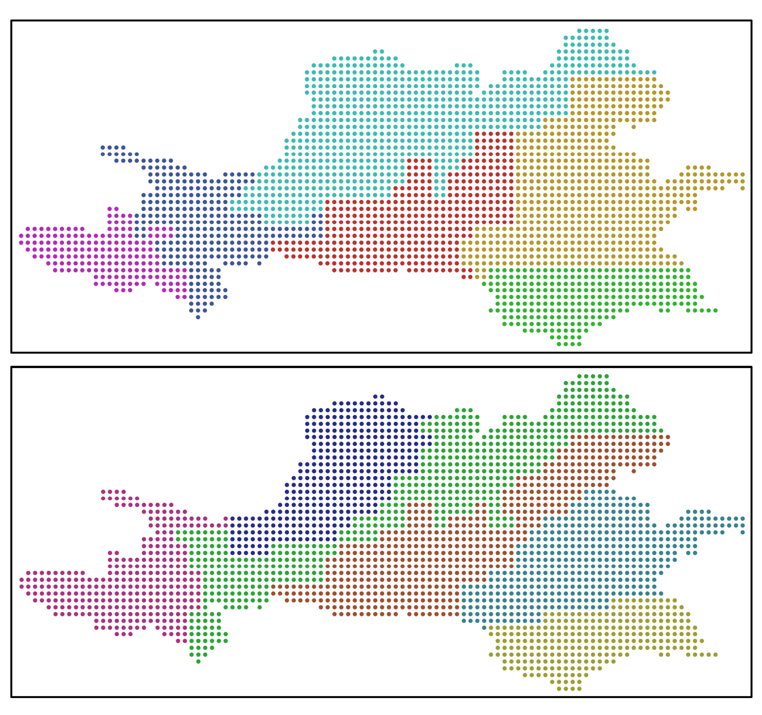


Fig. 6 The map of climate regions (Top: ArcGIS, bottom: ILS)
图 6 研究区气候区划示意图 (上图 ArcGIS，下图 ILS 算法)

5 结论

本文提出了一个改进 ILS 算法用于求解区划问题。该算法由初始解生成、局部搜索、群解搜索、解扰动、中心点更新等部分构成，且通过分区空间连续判断和修复操作保证当前解中所有分区空间连续。该算法基于分区中心点评价分区目标值，大幅降低了目标函数的计算，从而提升了算法效率。算法采用群搜索、解扰动和中心点更新，扩大了解的搜索空间，从而提升分区质量。基准案例测试表明：改进 ILS 算法区划结果明显优于 Skater 算法和 ARISEL 算法。对于无明显气候分区边界的多属性气候分区，改进 ILS 算法分区目标值显著优于 Skater 算法，分区结果与区域地形、气温、降雨的分布模式相吻合。

本文改进 ILS 算法设计具有几个显著的特点和优势。第一，与 AZP、AZP-SA、AZP-Tabu 和 ARISEL 相比，ILS 算法选择分区中心点进行目标函数计算，避免了局部搜索过程频繁地计算分区中单元属性均值，从而大幅地提高了算法计算效率。第二，AZP 算法属于简单的启发式算法，AZP-SA、AZP-Tabu 算法改进了搜索策略，属于元启发算法范畴，有效地提升了算法质量， ARISEL 使用多个初始解，并选择高质量解进行禁忌搜索，扩大了搜索空间。改进 ILS 算法使用群解、扰动、中心点更新等方法，区别有现有算法设计，充分利用了成熟的优化算法机制。第三，SKATER 仅考虑相邻单元区之间的相似性，大幅降低了搜索空间，计算效率很高。而 REDCAP 算法仅考虑相邻分区之间的相似性，通过自下而上方式完成聚类。ILS 算法通过搜索和扰动克服了 SKATER 和 REDCAP 算法的过于短视

的局限，有利于搜索到高质量分区。综上，ILS 算法的这些特征，保证了分区质量，又有效降低了算法的复杂度。

考虑到地理现象的空间渐变性、地理系统的复杂性、空间分异规律的尺度依赖性，本文区划算法的使用应建立在区域研究基础上：把握地理现象格局与变化机理，理解特定区域的地理特征，明确区划任务与目标，进而选择合适的区划指标。进一步的研究方向包括：如何确定合适的分区数量，如何进行数据标准化处理，如何选择最适宜差异度函数，以及如何基于本文算法发展出通用的区划方法和软件工具。

参考文献

[1] 郑度, 葛全胜, 张雪芹, 等. 中国区划工作的回顾与展望[J]. 地理研究, 2005, 24(3): 330-344.

[2] 刘燕华, 郑度, 葛全胜, 等. 关于开展中国综合区划研究若干问题的认识[J]. 地理研究, 2005, 24(3): 321-329.

[3] 郑度, 欧阳, 周成虎. 对自然地理区划方法的认识与思考[J]. 地理学报, 2008, 63(6): 563-573.

[4] Duque J C, Ramos R, Suriñach J. Supervised regionalization methods: A survey [J]. International Regional Science Review, 2007, 30(3): 195-220.

[5] Cliff A D, Haggett P, Ord J K, et al. Elements of Spatial Structure: A Quantitative Approach [M]. New York: Cambridge University Press, 1975.

[6] Keane M. The size of the region-building problem [J]. Environment and Planning A, 1975, 7(5): 575-577.

[7] Wright J, Revelle C, Cohon J. A multiobjective integer programming model for the land acquisition problem [J]. Regional Science and Urban Economics, 1983, 13(1): 31-53.

[8] Cova T J, Church R L. Contiguity constraints for single-region site search problems [J]. Geographical Analysis, 2000, 32(4): 306-329.

[9] Williams J C. A Zero-One Programming Model for Contiguous Land Acquisition [J]. Geographical Analysis, 2002, 34(4): 330-349.

[10] Shirabe T. A model of contiguity for spatial unit allocation [J]. Geographical Analysis, 2005, 37(1): 2-16.

[11] Duque J C, Church R L, Middleton R S. The p-Region Problem [J]. Geographical Analysis, 2011, 43,104–126.

[12] Li W, Church R L, Goodchild M F. The p-compact-regions problem [J]. Geographical Analysis, 2014, 46(3): 250-273.

[13] Guo D. Regionalization with dynamically constrained agglomerative clustering and partitioning (REDCAP) [J]. International Journal of Geographical Information Science, 2008, 22(7): 801-823.

[14] Openshaw S. A geographical solution to scale and aggregation problems in region-building, partitioning and spatial modelling [J]. Transactions of the Institute of British Geographers, 1977: 459-472.

[15] Browdy M H. Simulated annealing: an improved computer model for political redistricting [J]. Yale Law & Policy Review, 1990: 163-179.

[16] Openshaw S, Rao L. Algorithms for reengineering 1991 Census geography [J]. Environment

and planning A, 1995, 27(3): 425-446.
[17] Duque J C, Church R L. A new heuristic model for designing analytical regions[C]//North American Meeting of the International Regional Science Association, Seattle. 2004.
[18] 郭仁忠. 二维有序聚类方法及其在编制区划地图中的应用[J]. 武汉测绘学院学报, 1985, (2):21-29.
[19] Maravalle M, Simeone B. A spanning tree heuristic for regional clustering [J]. Communications in statistics-theory and methods, 1995, 24(3): 625-639.
[20] Assunção R M, Neves M C, Câmara G, et al. Efficient regionalization techniques for socio-economic geographical units using minimum spanning trees [J]. International Journal of Geographical Information Science, 2006, 20(7): 797-811.
[21] Aydin O, Janikas M V, Assunção R M, et al. A quantitative comparison of regionalization methods [J]. International Journal of Geographical Information Science, 2021, 35(11): 2287-2315.
[22] Lourenço H R, Martin O, Stützle T. Iterated Local Search: Framework and Applications [M] // Gendreau M, Potvin JY. eds. Handbook of Metaheuristics, 2nd. Edition. New York: Springer, 2010, 363-397.
[23] Xiao N. A unified conceptual framework for geographical optimization using evolutionary algorithms. Annals of the Association of American Geographers, 2008, 98(4)：795-817.
[24] Liu, Y., Cho, W. K., and Wang, S., 2016. PEAR: a massively parallel evolutionary computation approach for political redistricting optimization and analysis. Swarm and evolutionary computation, 30, 78-92.

An improved iterative local search algorithm for the regionalization problem

Yunfeng Kong

(Key Laboratory of Geospatial Technology for the Middle and Lower Yellow River Regions, Ministry of Education, Henan University, 47500)

Abstract: Regionalization is to divide a large geographic area into a number of homogenous and spatially contiguous regions. It has been widely used in fields such as geography, cartography, ecology, environment management, socio-economy, and urban planning. Since the general regionalization problem has been proven to be NP-Hard, various models and solution methods for regionalization have be proposed since 1960s. The regionalization methods can be classified into four categories: exact, clustering-based, heuristic, and tree-based. However, the commonly used regionalization algorithms are difficult to solve the problem in an effective and efficient manner simultaneously. An improved iterative local search algorithm is proposed in this paper for the regionalization problem. There are six key mechanisms in the new algorithm: the search of moving boundary units to improve the current solution; the center-based approach to accelerate the computation of solution objective; the solution perturbation to escape from the state of local optimum; the frequent update of regional centers to reevaluate the solution; the population-based

search to explore larger solution space; and the region repair to keep spatially contiguous regions. The regionalization experimentations on 55 benchmark instances show that the proposed algorithms outperforms ARISEL algorithm and SKATER algorithm in terms of sum-squared errors and adjusted Rand index. A case study of the climate regionalization using 60 attributes illustrates that the improved ILS is effective to delineate climate regions that are compatible with the precipitation, temperature and landform patterns.

Keywords: regionalization problem; iterative local search; benchmark test; case study